SONU KUMAR(RA2111030010021)

- **C files**
  - server.c
    - gcc server.c
    - ./a.out
  - station1.c
    - gcc station1.c
    - ./a.out 239.192.4.1
  - station2.c
    - gcc station2.c
    - ./a.out 239.192.4.2
  - client.c
    - gcc `pkg-config --cflags gtk+-3.0` -o client client.c `pkg-config --libs gtk+-3.0`
    - sudo ./client <IP-ADDRESS of the server>
  - receiver.c
    - Compiled and executed by the client.

- **Execution steps:**
  - ➔ First of all, client will send a join request to the server to join the multicast group.
  - ➔ After that Server will provide station list, site info to the client through TCP.
  - ➔ Then whichever station it selects from the station list, it is connected to that station.
  - ➔ All the stations are sending data, irrespective of client is connected or not. This functionality is incorporated to relate more with real life situation, e.g Tv/radio sends data even though there is no receiver connected.
  - ➔ Whenever receiver connects to a particular station, it starts receiving live-streaming videos from that station.
  - ➔ Used Media player: **ffplay**. All videos at station side is converted using ffmpeg to make it streamable.
    - ◆ Command used for conversion: ffmpeg -i inputfile.mp4 -f mpegts streamable_output.mp4
  - ➔ Receiver can pause, resume, change station or even terminate at any given time from GUI using thread.

    - ◆ **Pause** : It closes the multicast reception. This is implemented to relate with real life.

      When this button is pressed, it will generate an interrupt by changing flag value, which will temporarily stop receiving data from sender.

    - ◆ **Resume :** resumes it, keeping the station the same.

      When this button is pressed, it will generate an interrupt by again changing flag value, which will start receiving data from sender.

◆ **Change Station :** Receiver can change it anytime.

❑ Firstly it is disconnected from the station to whom it was connected earlier and then is connected to a new station as per receiver's choice and starts receiving respective live-streaming of data from that station.

◆ **Terminate :** Whenever it is selected, it is disconnected from the station it was connected earlier.

It will exit station, we have done this using 'pkill <media player>'

- Thread is used so that two processes can run in parallel:
  1) GUI
  2) Socket programming to send and receive the data

- **Design Configurations**

1. Client to Server: TCP
   a. TCP is used for one to one connection from client to server and it is used for station info and site info
2. Sender to Receiver: UDP
   a. UDP is used to send multicast live-streaming videos from sender to all receivers who joined multicast group.
3. Implementation of GUI: using **gtk**.
4. For all the previous functionalities, we have implemented four different functions which handles pause, resume, change the station, and Terminate accordingly.

- **Station Information**

  Station 1 name: F.R.I.E.N.D.S

  Station 2 name: H.I.M.Y.M

  Port Used for both stations: 5432

  Multicast Address for station 1: 239.192.4.1

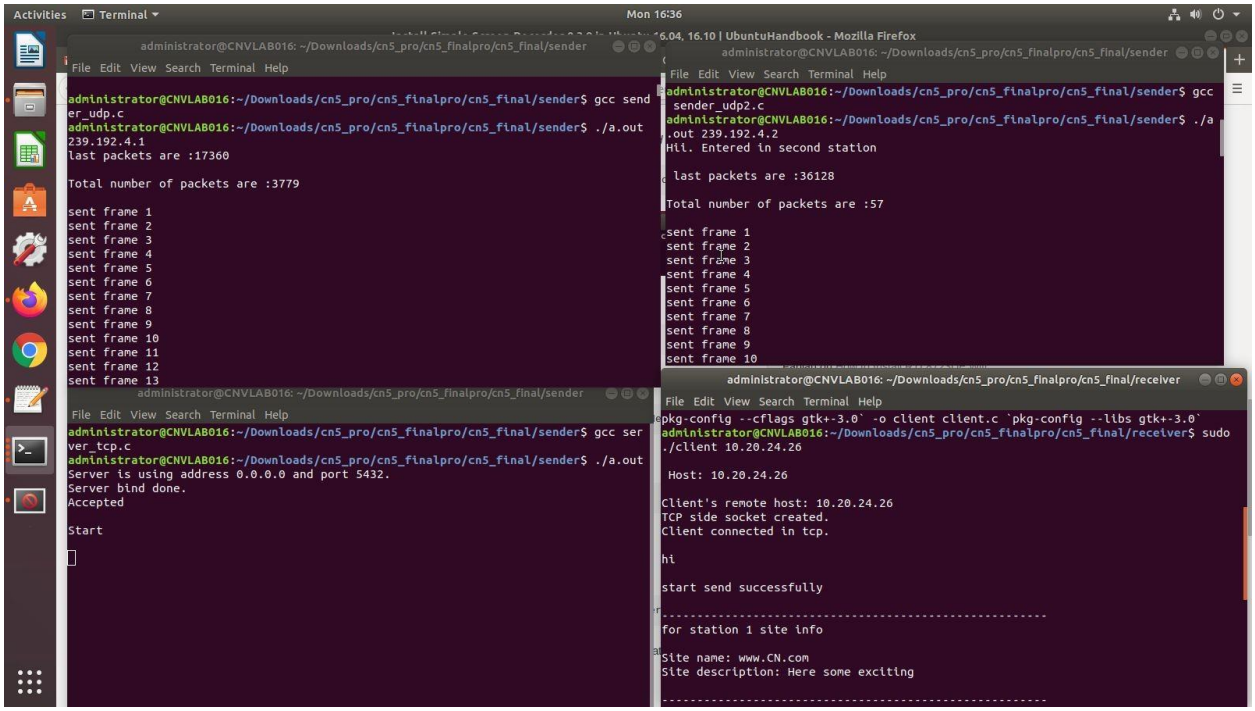
  Multicast Address for station 2: 239.192.4.2

- **Features:**
  - ❏ Receiver is receiving audio as well as video without any loss of data through UDP.
  - ❏ Both the stations are sending data on the same port, but having different IP addresses.

- **Buffer calculation:**
  - Time (t seconds) is initially declared and bit-rate will depend on station. Thus, the size of the buffer should be large enough to hold received data of t seconds.
  - Every time when the station is changed buffer-size will be recalculated according to t and bit-rate. For a particular station bit-rate is fixed and which is approximated.
  - By calculation, we got the buffer size = 64000.

- **Screen shots:**

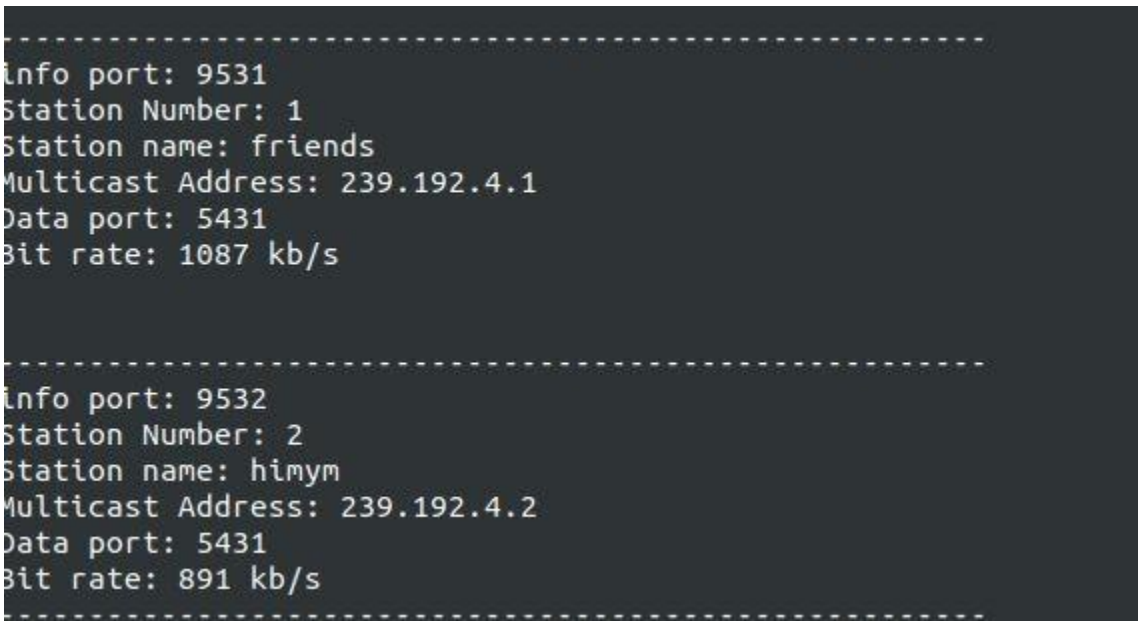
1. Successful compilation of all files.



2. Site info and station info at client side

Site info at client side

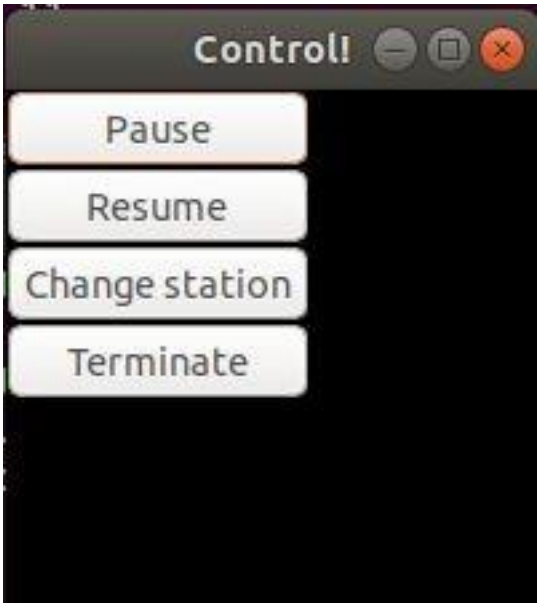
Station info at client side



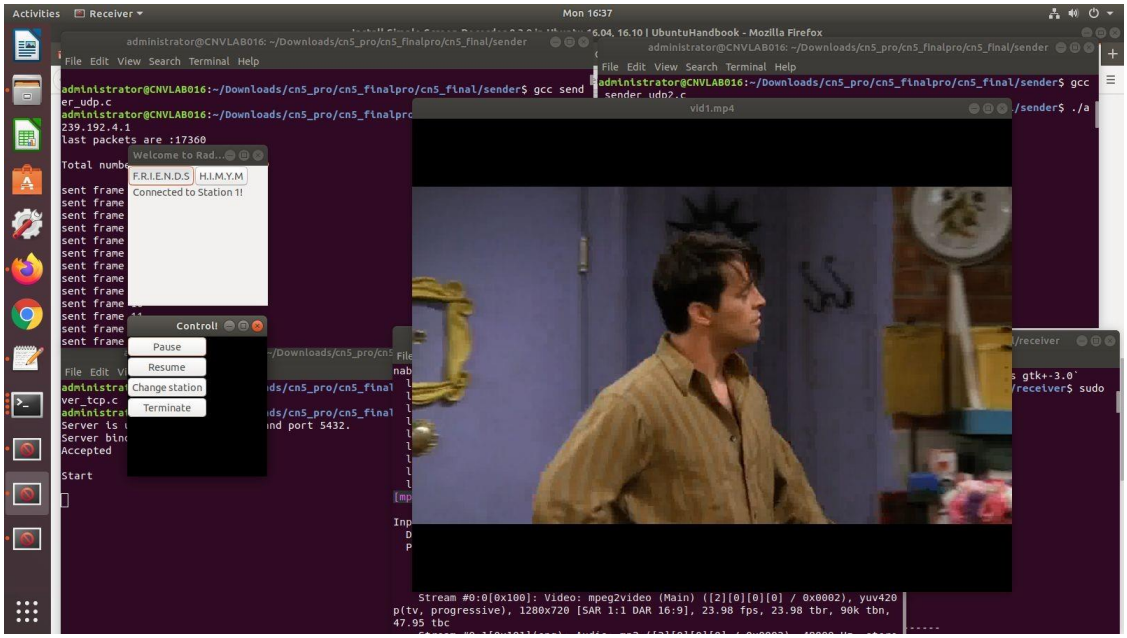
3. Station selection GUI window

➔ Page 1 opens for the client whenever he is connected to the server.

➔ Page 2 opens after selecting any of the two buttons : F.R.I.E.N.D.S or H.I.M.Y.M

Page 1 of GUI	Page 2 of GUI
	

4.Station 1 :F.R.I.E.N.D.S



5. Station 2: H.I.M.Y.M